Detailed Description

Referring to Figure 1, a functional block F may be controlled by a power-on reset circuit 10 to prevent its release for normal operation prior to the time that both the power supply has stabilized and the logic in the functional block F has transitioned to predetermined logic states. The power-on reset circuit 10, in one embodiment, may include the decision logic 12, a latch 14, a pulse generator 16, and an amplifier 18a that couples a signal feedback 20 back to the functional block F. The circuit 10 may be integrated on the same chip as the block F in one embodiment.

The functional block F may generate a plurality of output signals I_0 through I_N. Each of the signals is received by the decision logic 12 so that the decision logic 12 can determine whether the logic of the functional block F is in the proper predetermined, initial states to begin normal operation.

During start-up, the signals I_0 through I_N may be in some random state where it is highly probable that at least some of these signals are high and some are low. The decision logic 12 yields a low output to the S node of the latch 14 if one or more of the outputs I_0 to I_N of the functional block F is not in its predetermined state. The decision logic tests the signals I_0 to I_N to determine whether those signals are in their proper initial states.

Meanwhile, the pulse generator 16 initially generates a high pulse into the R node of the latch 14 when the supply voltage is ramping up. The combination of a low S node and high R node may result in a reset signal being sent to the functional block F through the amplifier 18a and the signal feedback 20. Thus, the pulse generator 16 may trigger the latch 14 to generate the reset signal to place the logic in the functional block into correct initial states.

The reset signal resets the logic in the functional block F to a desired predetermined state. As a result, the outputs I_0 through I_N yield known good states. When these good states are detected by the decision logic 12, this results in the latch 14 node S going high. The reset signal remains active until the node S has become high, indicating that the functional block's logic is ready, and the pulse generator 16 signal has gone away, indicating that the power supply is now fully operational.

When the pulse generator 16 signal is gone and logic indicates ready (S node = 1), normal chip operation begins in the functional block F. If the logic in the functional block F is not ready, for example due to long routing or for some other reason, the reset remains active, preventing normal operation of the function block F.

After the logic in the functional block F is released, the decision logic 12 may indicate a faulty ready state.

This may be the result of complex logic patterns or unexpected data patterns, as two examples. The reset signal is not improvidently activated since it is latched into a deactive state by the latch 14 until the power cycles. The latch 14 releases from its deactivated state (no reset) only if the pulse generator 16 indicates that the power supply is ramping.

Referring, next to Figure 2, the functional block F in this example is a counter having logic in the form of a plurality of flops only a few of which are shown. The reset signal (on feedback 20) to the flops is generated as soon as the pulse generator 16 is activated. The reset signal is active regardless of the indication from the logic within the functional block F. This is a desirable property because during start-up it is possible to have faulty indicators from the functional block F.

The feedback signal 20 resets the counter to a predetermined state. The counter sends an "all high" signal to the AND gate 12a (which acts as the decision logic) once all the flops are in their required logic states. Once the pulse generator 16 signal is gone (R=0), and the counter has indicated ready (S=1) the reset signal goes away. When the S and R inputs are zero or the R signal is one, the reset signal is active.

Potentially, the output of the AND gate 12a may be switched between low and high levels due to normal logic